IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results **BROWSE SEARCH IEEE GUID**

Results for "(((debugging cfg object annotate hierarchy)<in>metadata) >= 1990 <and...."
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((debugging cfg object annotate hierarchy)<in>met

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec

Google Scholar BETA

analyzer cfg annotate object | 1990 - 2

Scholar All articles Recent articles Results **1 - 10** of about **565** for **analyzer**

All Results
M Smith
J Choi
D Richardson
C Liu
G Holloway

A metrics-driven approach for utilizing concurrency in **object**-oriented real-time systems
LR Welch - ACM SIGPLAN OOPS Messenger, 1996 - portal.acm.org
... the statement-level control flow graph (**CFG**) by examining ... Figure 3: The **analysis** process ... arbitrary inter-**object** concurrency relations from **annotated** call graphs ...
Cited by 6 - Related Articles - Web Search - BL Direct

An Introduction to Machine SUIF and its Portable Libraries for Analysis and Optimization - group of 8 »
MD Smith, G Holloway - Division of Engineering and Applied Sciences, Harvard ..., 2001 - eecs.harvard.edu
... In particular, the peep pass uses the **CFG** and BVD ... The CFA library is used when dominator **analysis** is needed ... string specified in the target lib **annotation** on the ...
Cited by 53 - Related Articles - View as HTML - Web Search

Efficient online optimization by utilizing offline analysis and the safeTSA representation - group of 5 »
J von Ronne, A Hartmann, W Amme, M Franz - Proceedings of the inaugural conference on the Principles ..., 2002 - portal.acm.org
... exists a path through the program's **CFG** which starts ... to con- vey the results of escape **analysis** in order ... Nec97] and TAL [GMG99] are **annotation** techniques that ...
Cited by 5 - Related Articles - Web Search

A Novel Gain Time Reclaiming Framework Integrating WCET Analysis for **Object**-Oriented Real-Time ... - group of 3 »
EYS Hu, A Wellings, G Bernat - Second workshop on WCET analysis, 2002 - www-users.cs.york.ac.uk
... The OGTRG can be produced from the **analysis** of the OTLG and **CFG ...** Based on the **CFG**, an OTLG for each **object** or those **objects** de- noted with **annotation** A3 in **...**
Cited by 2 - Related Articles - View as HTML - Web Search

PLEIADES: an **object** management system for software engineering environments - group of 3 »
P Tarr, LA Clarke - Proceedings of the 1st ACM SIGSOFT symposium on Foundations ..., 1993 - portal.acm.org
... For example, a data flow **analysis** tool might be employed to detect anomalous sequences of events (38 **... CFG** and def/ref annotations **...** 3 **Object** Management Requirements **...**
Cited by 32 - Related Articles - Web Search - BL Direct

Partial Evaluation for Common Intermediate Language - group of 2 »
AM Chepovsky, AV Klimov, AV Klimov, YA Klimov, AS ... - Perspectives of System Informatics//Andrei Ershov Fifth ..., 2003 - Springer
... Graph (**CFG**). – Binding Time **Analysis** (BTA): CIL instructions in **CFG** are **annotated** as static, dynamic or special. To represent **...**
Cited by 2 - Related Articles - Web Search - BL Direct

Object-based data flow testing of web applications - group of 3 »
CH Liu, DC Kung, P Hsia - Quality Software, 2000. Proceedings. First Asia-Pacific ..., 2000 - ieeexplore.ieee.org
... In the past two decades, data flow **analysis** techniques

have been used to guide the
testing of traditional and **Object**-Oriented programs [S, 9, 10, 11, 121. **...**
Cited by 25 - Related Articles - Web Search

Automatic analysis of consistency between implementations andrequirements: a case study - group of 2 »
M Chechik, J Cannon - Computer Assurance, 1995. COMPASS'95.'Systems Integrity, ..., 1995 - ieeexplore.ieee.org
... 2.3 Improvements to **Analyzer** During this case study, we found out that we need a
richer **annotation** language and better **analysis** techniques than were **...**
Cited by 7 - Related Articles - Web Search

Escape analysis for Java - group of 15 »
JD Choi, M Gupta, M Serrano, VC Sreedhar, S ... - ... SIGPLAN conference on **Object**-oriented programming, systems, ..., 1999 - portal.acm.org
... M. The intuition easily extends to the escapement of an **object** from a thread. 3
lntraprocedural **Analysis** Given the control flow graph (**CFG**) representation of a **...**
Cited by 293 - Related Articles - Web Search - BL Direct

OpenJIT Frontend System: an implementation of the reflective JIT compiler frontend - group of 9 »
H Ogawa, K Shimura, S Matsuoka, F Maruyama, Y ... - LNCS 1826: Reflection and Software Engineering, 2000 - Springer
... published so far analyzed the **CFG** directly, and **...** The OpenJIT Class **Annotation Analyzer**
module extracts the **...** array assuming that the **annotation object** has been **...**
Cited by 5 - Related Articles - Web Search

Goooooooooogle ▶
Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

analyzer cfg annotate object Search

Google Home - About Google - About Google Scholar

©2007 Google

Enter Search Query Search

Home | Digital Library | Site Map | Store | Contact Us | Press Room | Shop

digital library

DIGITAL LIBRARY HOME

BROWSE BY TITLE

BROWSE BY SUBJECT

SEARCH

LIBRARY/INSTITUTION RESOURCES

RESOURCES

SUBSCRIPTION

ABOUT THE DIGITAL LIBRARY

rchive Page >> Table of Contents >> Abstr

000 IEEE International Symposium on V
anguages (VL'00) p. 5

ligh-Level Static and Dynamic Visualization of oftware Architectures

ohn Grundy, University of Auckland
ohn Hosking, University of Auckland

ull Article Text: PDF BUY ARTICLE IEEE XPLORE

OI Bookmark:

http://doi.ieeecomputersociety.org/10.1109/VL.200

Abstract

Developing software architectures for complex so applications is challenging, and requires good stat dynamic visualisation support. We describe the vi software architecture modelling visual language w developed and its support in the SoftArch environ Static software architecture views are developed u this language, and designs and implementations

developed from these specifications. Static views copied, animated and annotated to visualize runni system architecture characteristics. This approach provides better static modelling and dynamic visualisation of software architectures, at varying of abstraction, than do other current techniques.

Additional Information Back

Citation: John Grundy, John Hosking, "High-Le Static and Dynamic Visualization of Software Architectures," *vl*, p. 5, 2000 IEEE International Symposium on Visual Languages (VL'00), 2000.

Usage of this product signifies your acceptance of the Terms of Use.

This site and all contents (unless otherwise noted) are Copyright © 2000, IEEE, Inc. All rights reserved.

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Se
Search: The ACM Digital Library The
+visualizer +object +graph +annotate +cfg
Feedback Report a problem

Terms used **visualizer object graph annotate cfg**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanc
Try this search

Results 1 - 1 of 1

R

1 Visualization for program understanding: A system for graph-based visuali evolution of software
Christian Collberg, Stephen Kobourov, Jasvir Nagra, Jacob Pitts, Kevin W
June 2003 **Proceedings of the 2003 ACM symposium on Software visua**
Publisher: ACM Press
Full text available: pdf(3.80 MB) Additional Information: full citation, abst citings

We describe GEVOL, a system that visualizes the evolution of software drawing technique for visualization of large graphs with a temporal com extracts information about a Java program stored within a CVS version c displays it using a temporal graph visualizer. This information can be us to understand the evolution of a legacy program: Why is the program str is? Which programmers were responsibl ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machiner
ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contac

Useful downloads: Adobe Acrobat QuickTime Windows Med

Player

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Se
Search: ◉ The ACM Digital Library ○ The
+visual +debug +graph +annotate
Feedback Report a probl

Published since January 1990 and Published before September 2003
Terms used **visual debug graph annotate**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
☐ Open results in a new window
Try an Adv
Try this se

Results 1 - 20 of 73 Result page: **1** 2 3 4 next

1 Toward visual debugging: integrating algorithm animation capabilities witl debugger
Sougata Mukherjea, John T. Stasko
September 1994 **ACM Transactions on Computer-Human Interaction (** Issue 3
Publisher: ACM Press
Full text available: pdf(1.87 MB) Additional Information: full citation, ab index terms

Much of the recent research in software visualization has been polarized domains. In one domain that we call data structure and program visualiz views of program structures are generated automatically. These types of require programmer input or intervention, can be useful for testing and d however, their generic, low-level views are not expressive enough to con pro ...

Keywords: algorithm animation, debugging, programming environment user interfaces

2 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for A Collaborative research**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, ab terms

Understanding distributed applications is a tedious and difficult task. Vis process-time diagrams are often used to obtain a better understanding of application. The visualization tool we use is Poet, an event tracer develo Waterloo. However, these diagrams are often very complex and do not p desired overview of the application. In our experience, such tools displa non-trivial commun ...

3 Debugging and finding faults: Interactive visual debugging with UML
Timothy Jacobs, Benjamin Musial
June 2003 **Proceedings of the 2003 ACM symposium on Software visua**
Publisher: ACM Press
Full text available: pdf(264.18 KB) Additional Information: full citation, ab terms

Software debugging is an extremely difficult cognitive process requiring application behavior along with detailed understanding of specific applic Typical debuggers provide inadequate support for this process, focusing accessible through source code. To overcome this deficiency, we link dy state to a Unified Modeling Language (UML) object diagram. We enhan diagram with focus + context, gr ...

Keywords: Unified Modeling Language (UML), software visualization

4 Component-based software engineering: A component-based visual enviro process
G. Costagliola, R. Francese, M. Risi, G. Scanniello, A. De Lucia
July 2002 **Proceedings of the 14th international conference on Softwar knowledge engineering SEKE '02**
Publisher: ACM Press
Full text available: pdf(149.69 Additional Information: full citation, ab

KB)

We present the Component-Based Visual Environment Development (C] building visual language environments and introduce the Visual Langua supporting its implementation. The proposed approach is based on softw granularity levels and enables incremental development. The VLDesk ex gained from the development of the Visual Language Compiler-Compile functionalities with many adjunctive features use ...

Keywords: component development, development process, methodolog visual language

5 A visualization system for parallelizing programs
C.-R. Dow, S.-K. Chang, M. L. Soffa
December 1992 **Proceedings of the 1992 ACM/IEEE conference on Sup**
Publisher: IEEE Computer Society Press
Full text available: pdf(1.01 MB) Additional Information: full citation, re terms

6 A VISUAL ENVIRONMENT FOR DISTRIBUTED SIMULATION SYS
James H. Graham, Adel S. Elmaghraby, Irfan Karachiwala, Hussam Solim
January 1996 **ACM SIGSIM Simulation Digest**, Volume 25 Issue 3
Publisher: ACM Press
Full text available: pdf(919.47 KB) Additional Information: full citation, ab

Parallel and Distributed Simulation (PADS) algorithms are typically cate of two categories. They are either conservative or optimistic with respect handling causality. Conservative systems strictly preserve causality, whi detect and correct causality errors when they occur. *Time Warp* is the ba algorithms where rolling back the simulation clock allows the simulation Global Virtual Time ...

7 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JER**
Publisher: ACM Press

Full text available: pdf(613.63 KB) html (2.78 KB) Additional Information: full citation, re terms

8 Distributed environment: Narratives of space and time: visualization for di
Patrick J. Finnigan, Kelly A. Lyons
October 1991 **Proceedings of the 1991 conference of the Centre for Adv Collaborative research**
Publisher: IBM Press
Full text available: pdf(1.65 MB) Additional Information: full citation, ab

Programmers of distributed applications face the challenge of building c (*CP*) in a complex, heterogeneous network with distributed data and serv to build these systems are emerging, but widespread acceptance will req visualization and user interface technologies to reduce complexity.In thi and describe some initial results for visualizing three aspects of distribut [25) ...

Keywords: distributed systems, iconic programming, network managem

9 Technical papers: requirements engineering: Detection of conflicting funct case-driven approach: a static analysis technique based on graph transform
Jan Hendrik Hausmann, Reiko Heckel, Gabi Taentzer
May 2002 **Proceedings of the 24th International Conference on Softwa**
Publisher: ACM Press
Full text available: pdf(1.55 MB) Additional Information: full citation, ab index terms

In object-oriented software development, requirements of different stake manifested in use case models which complement the static domain mod functional requirements. In the course of development, these requiremen integrated to produce a consistent overall requirements specification. Ite be triggered by conflicts between requirements of different parties.Howe incompleteness, and informal nature ...

Keywords: UML, graph transformation, requirements specification, unil

10 VisualGraph: a graph class designed for both undergraduate students and e
Jeff Lucas, Thomas L. Naps, Guido Rossling
January 2003 **ACM SIGCSE Bulletin , Proceedings of the 34th SIGCSE Computer science education SIGCSE '03**, Volume 35 Issu
Publisher: ACM Press
Full text available: pdf(1.55 MB) Additional Information: full citation, ab terms

Graphs and graph algorithms play an important role in undergraduate dat algorithms courses. However, they often also represent the first case whe and the underlying concepts of the algorithms are not evident. Both stud therefore benefit from a simple yet expressive tool for coding graph algo conveniently visualizing them. We present such a tool, derived from a se requirements, and give an example applic ...

Keywords: CS 1, animation, graphs, pedagogy, visualization

11 A bug's eye view of immediate visual feedback in direct-manipulation prog
Curtis Cook, Margaret Burnett, Derrick Boom
October 1997 **Papers presented at the seventh workshop on Empirical s**
Publisher: ACM Press
Full text available: pdf(1.87 MB) Additional Information: full citation, re terms

12 An annotated bibliography of interactive program steering
Weiming Gu, Jeffrey Vetter, Karsten Schwan
September 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 9
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, ci

13 Limiting the probe effect in debugging concurrent object-oriented program
Ilene Seelemann
November 1995 **Proceedings of the 1995 conference of the Centre for A Collaborative research**
Publisher: IBM Press
Full text available: pdf(290.62 KB) Additional Information: full citation, ab terms

Event-based tracers for visualizing distributed applications use process-t demonstrating interaction among processes. Object-oriented programs ca similar presentation in which object-time diagrams are drawn and the int represents method invocations. In this type of diagram, it is necessary to methods involved.This paper presents an approach for resolving and stor names at debug time instead ...

14 Visualizing interactions in program executions
Dean F. Jerding, John T. Stasko, Thomas Ball
May 1997 **Proceedings of the 19th international conference on Softwar**
Publisher: ACM Press
Full text available: pdf(1.72 MB) Additional Information: full citation, re terms

Keywords: object-oriented software engineering, program understandin software visualization

15 Teaching computer graphics visual literacy to art and computer science stu resources and opportunities
Dena Eber, Rosalee Wolfe
May 2000 **ACM SIGGRAPH Computer Graphics**, Volume 34 Issue 2
Publisher: ACM Press
Full text available: pdf(943.87 KB) Additional Information: full citation, ab

Although it is a requisite skill for success in industry, visual literacy in g computer science and art students. Computer science majors are uneasy examine images while art students may not have much background in th

This column is the second in a two-part series that discusses an interdisci technique that overcomes these obstacles. Part one was published in *Con* February 2000, ...

16 Applying algorithm animation techniques for program tracing, debugging,
Sougata Mukherjea, John T. Stasko
May 1993 **Proceedings of the 15th international conference on Softwar**
Publisher: IEEE Computer Society Press
Full text available: pdf(909.31 KB) Additional Information: full citation, re

17 Finding and using implicit structure in human-organized spatial layouts of
Frank M. Shipman, Catherine C. Marshall, Thomas P. Moran
May 1995 **Proceedings of the SIGCHI conference on Human factors in**
Publisher: ACM Press/Addison-Wesley Publishing Co.
Full text available: html(39.44 KB) Additional Information: full citation, re terms

18 A test environment for natural language understanding systems
Li Li, Deborah A. Dahl, Lewis M. Norton, Marcia C. Linebarger, Dongdor
August 1998 **Proceedings of the 17th international conference on Com Volume 2 , Proceedings of the 36th annual meeting on As Computational Linguistics - Volume 2**
Publisher: Association for Computational Linguistics , Association for Co
Full text available: pdf(400.11 KB) Publisher Site Additional Information: full citation, ab

The Natural Language Understanding Engine Test Environment (ETE) i aids in the development and maintenance of large, modular, natural langı systems. Natural language understanding systems are composed of modu speech taggers, parsers and semantic analyzers) which are difficult to tes the complexity of their output data structures. Not only are the output da modules complex, b ...

19 Debugging temporal specifications with concept analysis
Glenn Ammons, David Mandelin, Rastislav Bodík, James R. Larus
May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA Programming language design and implementation PLDI '0**
Publisher: ACM Press
Full text available: pdf(154.43 KB) Additional Information: full citation, ab index terms

Program verification tools (such as model checkers and static analyzers) programs. These tools need formal specifications of correct program beh correct specification is difficult, just as writing a correct program is diffi methods for debugging programs, we need methods for debugging speci describes a novel method for debugging formal, temporal specifications. short program executi ...

Keywords: concept analysis, hierarchical clustering, specification debug specifications

20 A hybrid visual environment for models and objects
Paul A. Fishwick
December 1999 **Proceedings of the 31st conference on Winter simulatio to the future - Volume 2**
Publisher: ACM Press
Full text available: pdf(249.64 KB) Additional Information: full citation, re terms

Results 1 - 20 of 73 Result page: **1** 2 3 4 next

The ACM Portal is published by the Association for Computing Machinery Inc.
Terms of Usage Privacy Policy Code of Ethics Con

Useful downloads: Adobe Acrobat QuickTime Windows Medi

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Se
Search: ◉ The ACM Digital Library ○ The
+program +analysis +graph +annotate
Feedback Report a problem

Published since January 1990 and Published before September 2003
Terms used **program analysis graph annotate**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
☐ Open results in a new window
Try an Advanc
Try this search

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 ne
Best 200 shown R

1 Incremental analysis of constraint logic programs
Manuel Hermenegildo, German Puebla, Kim Marriott, Peter J. Stuckey
March 2000 **ACM Transactions on Programming Languages and Syste**
Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(399.84 KB) Additional Information: full citation, abst citings, index ter

Global analyzers traditionally read and analyze the entire program at onc nonincremental way. However, there are many situations which are not v simple model and which instead require reanalysis of certain parts of a p already been analyzed. In these cases, it appears inefficient to perform th program again from scratch, as needs to be done with current systems. W fixed-point algorithms used in current generic anal ...

Keywords: abstract interpretation, constraint logic programming, incren static analysis

2 Annotation-directed run-time specialization in C
Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, Susan J. E

December 1997 **ACM SIGPLAN Notices , Proceedings of the 1997 ACM symposium on Partial evaluation and semantics-based manipulation PEPM '97**, Volume 32 Issue 12

Publisher: ACM Press

Full text available: pdf(1.99 MB) Additional Information: full citation, abst citings, index ter

We present the design of a dynamic compilation system for C. Directed l user annotations specifying where and on what dynamic compilation is t binding time analysis computes the set of run-time constants at each pro annotated procedure's control flow graph; the analysis supports program· polyvariant division and specialization. The analysis results guide the co specialized run-time specializer for each dynamically c ...

3 Multi-pass execution of functional logic programs

Jukka Paakki

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symp Principles of programming languages**

Publisher: ACM Press

Full text available: pdf(1.33 MB) Additional Information: full citation, abst index terms

An operational semantics for functional logic programs is presented. In s functional terms provide for reduction of expressions, provided that they semantics is based on multi-pass evaluation techniques originally develo grammars. Program execution is divided into two phases: (1) constructio proof tree, and (2) its decoration into a complete proof tree. The construc modified SLD-resolution scheme, and the decorat ...

4 Embedded program timing analysis based on path clustering and architectu

R. Ernst, W. Ye

November 1997 **Proceedings of the 1997 IEEE/ACM international conf Computer-aided design**

Publisher: IEEE Computer Society

Full text available: pdf(315.03 KB) Publisher Site Additional Information: full citation, abst citings, index ter

Formal Program running time verification is an important issue in systen for performance optimization under "first-time-right" design constraints system verification. Simulation based approaches or simple instruction c appropriate and risky for more complex architectures in particular with d execution paths. Formal analysis techniques have suffered from loose tir to significant performance penalties when strictly adh ...

5 Points-to analysis for Java using annotated constraints

Atanas Rountev, Ana Milanova, Barbara G. Ryder

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM S conference on Object oriented programming, systems, la applications OOPSLA '01**, Volume 36 Issue 11

Publisher: ACM Press

Full text available: pdf(263.51 KB) Additional Information: full citation, abst citings, index ter

The goal of point-to analysis for Java is to determine the set of objects p reference variable or a reference object field. This information has a wid applications in optimizing compilers and software engineering tools. In t present a point-to analysis for Java based on Andersen's point-to analysi implement the analysis by using a constraint-based approach which emp *inclusion constraints.* Constraint annotations allow u ...

6 Pointer analysis for multithreaded programs

Radu Rugina, Martin Rinard

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 5

Publisher: ACM Press

Full text available: pdf(1.82 MB) Additional Information: full citation, abst citings, index ter

This paper presents a novel interprocedural, flow-sensitive, and context- analysis algorithm for multithreaded programs that may concurrently up For each pointer and each program point, the algorithm computes a cons approximation of the memory locations to which that pointer may point. correctly handles a full range of constructs in multithreaded programs, ir functions, function pointers, structures, arrays, nested stru ...

7 Pointer analysis for structured parallel programs
Radu Rugina, Martin C. Rinard
January 2003 **ACM Transactions on Programming Languages and Syst**
Volume 25 Issue 1
Publisher: ACM Press
Full text available: pdf(383.29 KB) Additional Information: full citation, abst index terms

This paper presents a novel interprocedural, flow-sensitive, and context-analysis algorithm for multithreaded programs that may concurrently up The algorithm is designed to handle programs with structured parallel co fork-join constructs, parallel loops, and conditionally spawned threads. F each program point, the algorithm computes a conservative approximati locations to which that pointer may point. Th ...

Keywords: Pointer analysis

8 Incremental global reoptimization of programs
Lori L. Pollock, Mary Lou Soffa
April 1992 **ACM Transactions on Programming Languages and Systen**
Volume 14 Issue 2
Publisher: ACM Press
Full text available: pdf(1.88 MB) Additional Information: full citation, abst citings, index ter

Although optimizing compilers have been quite successful in producing factors that limit their usefulness are the accompanying long compilatior of good symbolic debuggers for optimized code. One approach to attaini recompilations is to reduce the redundant analysis that is performed for c response to edits, and in particulars, small maintenance changes, without quality of the generated code. Although modular program ...

Keywords: compiler optimization, incremental data flow analysis, incre reoptimization, optimization dependencies

9 Alias annotations for program understanding
Jonathan Aldrich, Valentin Kostadinov, Craig Chambers

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM conference on Object-oriented programming, systems, applications OOPSLA '02**, Volume 37 Issue 11

Publisher: ACM Press

Full text available: pdf(336.14 KB) Additional Information: full citation, abst citings, index ter

One of the primary challenges in building and evolving large object-orie understanding aliasing between objects. Unexpected aliasing can lead to mistaken assumptions, security holes, and surprising side effects, all of v software defects and complicate software evolution.This paper presents capability-based alias annotation system for Java that makes alias patter source code, enabling developers to reason more effec ...

Keywords: aliasing, aliasjava, encapsulation, java, ownership types, typ uniqueness

10 Interactive type analysis and extended message splitting; optimizing dynan oriented programs

Craig Chambers, David Ungar

June 1990 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 6

Publisher: ACM Press

Full text available: pdf(1.58 MB) Additional Information: full citation, abst citings, index ter

Object-oriented languages have suffered from poor performance caused slow dynamically-bound procedure calls. The best way to speed up a prc compile it out, but dynamic binding of object-oriented procedure calls w type information precludes inlining. Iterative type analysis and extended are new compilation techniques that extract much of the necessary type i make it possib ...

11 Separation constraint partitioning: a new algorithm for partitioning non-str sequential threads

Klaus E. Schauser, David E. Culler, Seth C. Goldstein

January 1995 **Proceedings of the 22nd ACM SIGPLAN-SIGACT symp**

Principles of programming languages
Publisher: ACM Press
Full text available: pdf(1.56 MB) Additional Information: full citation, abst citings, index ter

In this paper we present substantially improved thread partitioning algor implicitly parallel languages. We present a new block partitioning algori constraint partitioning, which is both more powerful and more flexible th algorithms. Our algorithm is guaranteed to derive maximal threads. We p framework for proving the correctness of our partitioning approach, and separation constraint partitioning mak ...

12 Live-structure dataflow analysis for Prolog
Anne Mulkers, William Winsborough, Maurice Bruynooghe
March 1994 **ACM Transactions on Programming Languages and Syste** Volume 16 Issue 2
Publisher: ACM Press
Full text available: pdf(3.59 MB) Additional Information: full citation, abst citings, index ter

For the class of applicative programming languages, efficient methods fo memory occupied by released data structures constitute an important asp implementations. The present article addresses the problem of memory r programs through program analysis rather than by run-time garbage coll derive run-time properties that can be used at compile time to specialize program according to a given set of queries and ...

Keywords: Prolog, abstract interpretation, compile-time garbage collect program analysis

13 Parametric shape analysis via 3-valued logic
Mooly Sagiv, Thomas Reps, Reinhard Wilhelm
May 2002 **ACM Transactions on Programming Languages and System** Volume 24 Issue 3
Publisher: ACM Press
Full text available: pdf(1.10 MB) Additional Information: full citation, abst citings, index ter

Shape analysis concerns the problem of determining "shape invariants" f perform destructive updating on dynamically allocated storage. This arti parametric framework for shape analysis that can be instantiated in diffe different shape-analysis algorithms that provide varying degrees of effici A key innovation of the work is that the stores that can possibly arise du represented (conservatively) using 3-valued l ...

Keywords: 3-valued logic, Abstract interpretation, alias analysis, constr destructive updating, pointer analysis, shape analysis, static analysis

14 Improving the accuracy of Petri net-based analysis of concurrent programs

A. T. Chamillard, Lori A. Clarke

May 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings c SIGSOFT international symposium on Software testing and '96**, Volume 21 Issue 3

Publisher: ACM Press

Full text available: pdf(1.43 MB) Additional Information: full citation, abst citings, index ter

Spurious results are an inherent problem of most static analysis methods an effort to produce conservative results, overestimate the executable bel Infeasible paths and imprecise alias resolution are the two causes of such this paper we present an approach for improving the accuracy of Petri ne concurrent programs by including additional program state information i present empirical results that demonstr ...

15 Contextual def-use associations for object aggregation

Amie L. Souter, Lori L. Pollock

June 2001 **Proceedings of the 2001 ACM SIGPLAN-SIGSOFT worksh analysis for software tools and engineering**

Publisher: ACM Press

Full text available: pdf(178.90 KB) Additional Information: full citation, abst index terms

This paper presents a novel formulation of definitions, uses, and def-use objects in object-oriented programs by exploiting the relations that occu and their instantiated objects due to aggregation. Contextual def-use assc computed by generating a partial call sequence for each def and use base

aggregation relations. By extending an escape points-to graph representa we have developed and implemented three strategi ...

16 Type-base flow analysis: from polymorphic subtyping to CFL-reachability
Jakob Rehof, Manuel Fähndrich
January 2001 **ACM SIGPLAN Notices , Proceedings of the 28th ACM S SIGACT symposium on Principles of programming langı** Volume 36 Issue 3
Publisher: ACM Press
Full text available: pdf(1.23 MB) Additional Information: full citation, abst terms

We present a novel approach to scalable implementation of type-based fl polymorphic subtyping. Using a new presentation of polymorphic subyt instantiation constraints, we are able to apply context-free language (CFl techniques to type-based flow analysis. We develop a CFL-based algorit flow-information in time O(*n*³), where *n* is the size of the typed pr algorithm substantially improves upon the best previously ...

17 Automatic performance prediction to support cross development of paralle
Matthias Schumann
January 1996 **Proceedings of the SIGMETRICS symposium on Parallel tools**
Publisher: ACM Press
Full text available: pdf(1.32 MB) Additional Information: full citation, refe

18 Efficient composite data flow analysis applied to concurrent programs
Gleb Naumovich, Lori A. Clarke, Leon J. Osterweil
July 1998 **ACM SIGPLAN Notices , Proceedings of the 1998 ACM SIG workshop on Program analysis for software tools and engine** Volume 33 Issue 7
Publisher: ACM Press
Full text available: pdf(1.05 MB) Additional Information: full citation, abst citings, index ter

FLAVERS, a tool for verifying properties of concurrent systems, uses cc

analysis to incrementally improve the precision of the results of its verifi FLAVERS is one of the few static analysis techniques for concurrent sys potential to handle large scale systems, it sometimes can still be very exp this paper we experimentally compare the cost of two versions of this ap composite data flow analysis problems. The f ...

19 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V.

July 2001 **ACM Transactions on Programming Languages and System** Volume 23 Issue 4

Publisher: ACM Press

Full text available: pdf(1.95 MB) Additional Information: full citation, abst citings, index ter

Since the early days of logic programming, researchers in the field realiz exploitation of parallelism present in the execution of logic programs. Tl nature, the presence of nondeterminism, and their referential transparenc characteristics, make logic programs interesting candidates for obtaining parallel execution. At the same time, the fact that the typical applications programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic pr parallelism, prolog

20 Beyond ASIS: program data bases and tool-oriented queries

Janusz Laski, William Stanley, Pawel Podgorski

September 2001 **ACM SIGAda Ada Letters , Proceedings of the 2001 a SIGAda international conference on Ada SIGAda '01,** 4

Publisher: ACM Press

Full text available: pdf(49.44 KB) Additional Information: full citation, abst index terms

The availability of higher level ASIS libraries is of prime importance for ASIS technology to facilitate the development of Software Analysis and tools. This is due to the fact that ASIS queries are expressed in terms of immensely complex language and do not directly support the objectives this paper we discuss two plausible sets of higher levels, tool-oriented qu

Program Under Analysis (PUA), which ideally d ...

Keywords: ASIS, Ada, Software, dynamic analysis, program data bases dependencies, queries, static analysis, testing, verification

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 1

The ACM Portal is published by the Association for Computing Machinery
ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact

Useful downloads: Adobe Acrobat QuickTime Windows Med Player

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	867	717/124.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:28
L2	293	717/125.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:28
L3	425	717/131.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:28
L4	103	717/132.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:28
L5	79	717/133.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:29
L6	1240	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:32
L7	170	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack) and 717/???.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:33
L8	163	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack) and 717/???.ccls. and (render$3 or display$3 or present$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:33

EAST Search History

L9	145	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack) and 717/???.ccls. and (render$3 or display$3 or present$5) and (profil$3 or debug$4 or analy$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:35
L10	112	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack) and 717/???.ccls. and (render$3 or display$3 or present$5) and (profil$3 or debug$4 or analy$4) and ("control flow" or invocation or allocation or "object creation" or instantiation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:35
L11	82	(ide or visual$7 or analy$4) and (graph or tree or nodes) and (annotat$5) and (object near2 (contains or nest$3 or refer$7)) and (cfg or "control flow" or call$3 or stack) and 717/???.ccls. and (render$3 or display$3 or present$5) and (profil$3 or debug$4 or analy$4) and ("control flow" or invocation or allocation or "object creation" or instantiation) and "source code"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:50
L12	8	("6343376" "5313616" "6823507" "6381735" "6823507").pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/03 14:46
L13	28	("4533997" | "4931928" | "5263162" | "5432942" | "5590330" | "5793374" | "5854924" | "5862382" | "5881290" | "5933635" | "5937190" | "6009256" | "6071317" | "6125439" | "6151701" | "6154876" | "6240376" | "6243848" | "6311327" | "6336087" | "6412106" | "6594761" | "6601235" | "6766481" | "6779114" | "6820256" | "6928638" | "6961925").PN. OR ("7051322").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/01/03 15:13

EAST Search History

L14	20	(("4533997" | "4931928" | "5263162" | "5432942" | "5590330" | "5793374" | "5854924" | "5862382" | "5881290" | "5933635" | "5937190" | "6009256" | "6071317" | "6125439" | "6151701" | "6154876" | "6240376" | "6243848" | "6311327" | "6336087" | "6412106" | "6594761" | "6601235" | "6766481" | "6779114" | "6820256" | "6928638" | "6961925").PN. OR ("7051322").URPN.) and (tree or graph$3 or node)	US-PGPUB; USPAT; USOCR	OR	ON	2007/01/03 15:35
L15	892	(717/12?.ccls. or 717/13?.ccls.) and (graph$3 or tree or node) and (cfg or flow or call$3) and (object near3 (nest$3 or refer$3 or referenc$3 or contain$3 or hierarchy or hierarchical$3))	US-PGPUB; USPAT; USOCR	OR	ON	2007/01/03 15:41
L16	436	(717/12?.ccls. or 717/13?.ccls.) and (graph$3 or tree or node) same (cfg or flow or call$3) and (object near3 (nest$3 or refer$3 or referenc$3 or contain$3 or hierarchy or hierarchical$3))	US-PGPUB; USPAT; USOCR	OR	ON	2007/01/03 15:41
L17	11	(717/12?.ccls. or 717/13?.ccls.) and (graph$3 or tree or node) same (cfg or flow or call$3) same annotat$5 and (object near3 (nest$3 or refer$3 or referenc$3 or contain$3 or hierarchy or hierarchical$3)) and (gui or ui or visual$7 or view$3)	US-PGPUB; USPAT; USOCR	OR	ON	2007/01/03 15:42

Google Scholar BETA

determine object hierarchy flow cfg annotated | 1990 | - | 2

Scholar **All articles** Recent articles Results **1** - **10** of about **77** for **determine**

All Results
S Moon
A Krishnaswamy
D Richardson
M Mock
N Bellas

[PS] Program Slicing: An Application of **Object**-Oriented Program Dependency Graphs - group of 5 »
A Krishnaswamy - 1994 - korson-mcgregor.com
... oriented slice is more complex than **determining** either an ... features are seen as we move up the **hierarchy**. ... **Objects** interact in an **object**-oriented system by ...
Cited by 29 - Related Articles - View as HTML - Web Search

Automated maintenance of avionics software - group of 3 »
JP Loyall, SA Mathisen, PJ Hurley, JS Williamson - Aerospace and Electronics Conference, 1993. NAECON 1993., ..., 1993 - ieeexplore.ieee.org
... examines the coverage information produced to **determine** whether any ... the user to organize a **hierarchy** of test ... Multiple files of these **objects** can be maintained ...
Cited by 3 - Related Articles - Web Search - BL Direct

Metrics for quality and concurrency in **object**-based systems
LR Welch, M Lankala, W Farr, DK Hammer - Annals of Software Engineering, 1996 - Springer
... This section presents techniques to **determine** whether two statements can ... same **object** instance, ie, [(Si, Sj) 6 IDG(a ... 3. The control **flow** of method a is such ...
Cited by 10 - Related Articles - Web Search - BL Direct

OBJECT MANAGEMENT SUPPORT FOR THE CONSTRUCTION OF COMPLEX APPLICATIONS - group of 2 »
PL TARR - 1996 - laser.cs.umass.edu
... Control **Flow** Graph ... tion to **determine** which procedures would be aected if a ... suggests a number of necessary **object** management capabilities, as described below. ...
Cited by 1 - Related Articles - View as HTML - Web Search - Library Search

Developing and integrating ProDAG in the Arcadia environment - group of 3 »
DJ Richardson, TO O'Malley, CT Moore, SL Aha - ACM SIGSOFT Software Engineering Notes, 1992 - portal.acm.org
... ing, **object** management, user interface development, process ... loop by **determining** whether the loop terminates — that is, ... a DGI to the control **flow** graph and ...
Cited by 28 - Related Articles - Web Search - BL Direct

Applying predication to efficiently handle runtime class testing - group of 3 »
C Sadler, SKS Gupta, R Bhatia - ACM SIGARCH Computer Architecture News, 2000 - portal.acm.org
... some identifier(s) from the receiver **object** that can be ... 2. Evaluate each class test, and **determine** whether to ... in and performance of the memory **hierarchy** can be ...
Related Articles - Web Search - BL Direct

[PS] Virtual Method Resolution with Typed Alias Graphs - group of 5 »
M Schordan, W Amme - Proceedings of the 8th International Workshop on Compilers ..., 2000 - 143.205.180.128

... They apply type **hierarchy** analysis, type propagation, aggregate ... time alias analysis for **object**-oriented languages ... graphs we are able to **determine** which virtual ...
Related Articles - View as HTML - Web Search

An Efficient Resource-constrained Global Scheduling Technique For Superscalar And Vliw Processors - group of 2 »
SM Moon, K Ebcioglu - Microarchitecture, 1992. MICRO 25., Proceedings of the 25th ..., 1992 - ieeexplore.ieee.org
... 2.1 VLIW tree instruction The VLIW program is a control **flow** graph (**CFG**) where each node is ... Each directed edge of the tree is **annotated** with zero or more ...
Cited by 115 - Related Articles - Web Search - Library Search - BL Direct

Symbolic computing, Lisp languages, and parallel computing
MM Furnari, A Massarotti - Massively Parallel Computing Systems, 1994., Proceedings of ..., 1994 - ieeexplore.ieee.org
... parallelism is expressed by the task **hierarchy**, and the ... is generally captured into time Control **Flow** Graph (**CFG** ... y be two distinct nodes of the **CFG**, then exactly ...
Related Articles - Web Search

Program Comprehension - group of 7 »
S Rugaber - Encyclopedia of Computer Science and Technology, 1995 - cs.concordia.ca
... an AST by walking the tree to **determine** basic blocks ... In a PDG control and **data flow** dependencies are treated together ... oth- erwise require both a DFG and a **CFG**. ...
Cited by 23 - Related Articles - View as HTML - Web

Search

Gooooooooogle ▶

Result Page: **1** 2 3 4 5 6 7 8 **Next**

determine object hierarchy flow cfg | Search

Google Home - About Google - About Google Scholar

©2007 Google

Google Scholar BETA visual debugger object graph annotation 1990 - 2

Scholar All articles Recent articles Results **11 - 20** of about **833** for **visual d**

All Results
J Stasko
S Mukherjea
R Helm
J Vlissides
W De Pauw

Visual parallel programming with Visper - group of 4 »
N Stankovic, K Zhang - Proceedings of the High-Performance Computing on the ..., 1997 - doi.ieeecs.org
... deals with problems of logical and performance **debugging**. ... lines, that are drawn first during **visual** programming ... The tool is built upon an **object**-oriented model ...
Web Search

Visualizing Interactions in Program Executions - group of 11 »
DF Jerding, JT Stasko, T Ball - Software Engineering, 1997., Proceedings of the 1997 (19th) ..., 1997 - ieeexplore.ieee.org
... an example call trace, call tree, and call **graph**. ... Polka is an **object**-oriented toolkit for creating algorithm ... The major **visual** innovation in the Execution Mural ...
Cited by 89 - Related Articles - Web Search - Library Search - BL Direct

[BOOK] Object-oriented software construction - group of 23 »
B Meyer - 1997 - Prentice-Hall, Inc. Upper Saddle River, NJ, USA
... design of an OCL query-based **debugger** for C++ ... Jens Palsberg , Michael I. Schwartzbach, **Object**-oriented type ... run-time systems and its **visual** programming language ...
Cited by 3856 - Related Articles - Web Search - Library Search

Visual testing of software - group of 2 »
J Lönnberg - Master's thesis, Helsinki University of Technology, Oct, 2003 - cs.hut.fi
... 35 4.5.11 Slicing and dependence **graphs** . . ... procedural languages (with or without
object-orientation) such ... **visual** debuggers" (eg the Javix **Visual Debugger** [68 ...
Cited by 2 - Related Articles - View as HTML - Web Search

Complete visualizations of concurrent programs and their executions
KM Kahn, VA Saraswat - **Visual** Languages, 1990., Proceedings of the 1990 IEEE ..., 1990 - ieeexplore.ieee.org

... manner to facilitate the perception of **object** coherence and ... edit, under- stand, think
of, run, **debug** and otherwise ... patterns in static and dynamic **visual** input. ...
Cited by 102 - Related Articles - Web Search

Cover yourself with Skin - group of 7 »
JG Hosking, S Fenwick, WB Mugridge, JC Grundy - Proceedings of OZCHI'95, 1995 - itee.uq.edu.au
... Future work includes reuse of Skin **graphs** for visualising execution of Skin functions ...
[6] Fenwick, S. 1994: A **Visual Debugger** for **Object**-Oriented Programs ...
Cited by 8 - Related Articles - View as HTML - Web Search

An open **graph** visualization system and its applications to software engineering - group of 16 »
ER Gansner, SC North - Software Practice and Experience, 2000 - doi.wiley.com
... same as conventional scripting languages (**Visual** Basic, Unix ... Grappa has classes for
graph representation, presentation, and ... sub-classes of

Grappa **object** classes ...
Cited by 242 - Related Articles - Web Search - BL Direct

Design, construction, and application of a generic **visual** language generation environment - group of 8 »
K Zhang, DQ Zhang, J Cao - IEEE Transactions on Software Engineering, 2001 - doi.ieeecs.org
... as a compiler, its associated syntax directed editor, and **debugger**. ... 22 shows the **visual** objects used in a ... An **annotation object** can specify the **annotation** for a ...
Cited by 29 - Related Articles - Web Search - BL Direct

Event **graph** visualization for debugging large applications - group of 3 »
D Kranzlmüller, S Grabner, J Volkert - Proceedings of the SIGMETRICS symposium on Parallel and ..., 1996 - portal.acm.org
... This **object** oriented approach allows us to easily extend and ... The most important events in **debug**- ging message passing ... is also applied to the **visual**- ization of ...
Cited by 24 - Related Articles - Web Search

A component-based **visual** environment development process
G Costagliola, R Francese, M Risi, G Scanniello, A ... - Proceedings of the 14th international conference on Software ..., 2002 - portal.acm.org
... and represent the dynamic aspects of the **object** belonging to a ... **debugger** extends the classic definition of compiler **debugger** to the **visual** environment case. ...
Cited by 5 - Related Articles - Web Search

◄ Gooooooooo ogle ►
Result Page: **Previous** 1 **2** 3 4 5 6 7 8 9 10 11 **Next**

visual debugger object graph annota Search

Google Home - About Google - About Google Scholar

©2007 Google

Google Scholar BETA

object hierarchy flow hierarchy annotated ana | 1990 | - | 2

Scholar **All articles** Recent articles Results **1 - 10** of about **544** for **object** **hi**

All Results
C Chambers
J Ning
W Kozaczynski
F Wolf
D Ungar

A Knowledge Base for Program Debugging - group of 5 »
A Tubaishat - Proceedings of the ACS/IEEE International Conference on ..., 2001 - doi.ieeecomputersociety.org
... on B if there exist a control **flow** path from ... Their relationships in the knowledge **hierarchy** are indicated using ... format: If Chunk A MATCH Knowledge **Object** B has ...
Cited by 2 - Related Articles - Web Search

[PS] Program Slicing: An Application of **Object**-Oriented Program Dependency Graphs - group of 5 »
A Krishnaswamy - 1994 - korson-mcgregor.com
... **Dependence** Subgraph (CDS), a Control **Flow** Graph (CFG ... the representation is the Class **Hierarchy** Subgraph (CHS ... the three subgraphs is the **Object**-oriented Program ...
Cited by 29 - Related Articles - View as HTML - Web Search

A Knowledge-based Approach To Software System Understanding
W Kozaczynski, S Letovsky, J Ning, A Consulting - Knowledge-Based Software Engineering Conference, 1991. ..., 1991 - ieeexplore.ieee.org
... Once the ASTs are **annotated** with semantic descriptions ... but can be applied to non-**object**-oriented languages ... automatic program decomposition based on **flow** and **data** ...
Cited by 10 - Related Articles - Web Search

Issues in the testing of **object**-oriented software
EV Berard - Electro/94 International. Conference Proceedings. Combined ..., 1994 - ieeexplore.ieee.org
... Therefore, in addition to source code and **object** code, we ... exceptions, interrupts
cause changes in control **flow** without the ... Boundary value **analysis**," one of ...
Cited by 15 - Related Articles - Web Search - BL Direct

Meta-**Data** Components in Support of an Active Deductive **Object**-Oriented Database System - group of 3 »
TB Abdellatif, HWR Chan, SW Dietrich, B ... - Proc. Third IEEE Meta-**Data** Conf., Apr, 1999 - computer.org
... the root of the meta-model class **hierarchy** to require ... SW, and Urban, SD, "On Control
Flow Testing of Active Rules in a Declarative **Object**-Oriented Framework ...
Cited by 3 - Related Articles - Cached - Web Search

A prototype tool for **flow analysis** of **object**-oriented programs - group of 5 »
J Gustafsson - **Object**-Oriented Real-Time Distributed Computing, 2002.(ISORC ..., 2002 - ieeexplore.ieee.org
... time for **object**-oriented programs one needs **flow** information, such ... Figure 5. Class
hierarchy for example 3 ... The program creates a polymorphic **object** a. Since the ...
Cited by 3 - Related Articles - Web Search

Path clustering in software timing **analysis** - group of 5 »
FE Wolf, RW Ye - Very Large Scale Integration (VLSI) Systems, IEEE ..., 2001 - ieeexplore.ieee.org
... If adjacent PCS on one level of **hierarchy** or child PCS are classified as SFP, they
are ... The control **flow** defined by the OAM mode has been **annotated** using a ...
Cited by 11 - Related Articles - Web Search - BL Direct

Monitoring compliance of a software system with its high-level design models - group of 8 »
M Sefika, A Sane, RH Campbell - Proceedings ICSE, 1996 - doi.ieeecomputersociety.org
... oriented systems is an inheritance **hierarchy** rooted at ... static call graphs and animated **object** interactions, and ... of the relations invokes, control-**flow**, and datu ...
Cited by 83 - Related Articles - Web Search - BL Direct

Interactive visual debugging with UML
T Jacobs, B Musial - Proceedings of the 2003 ACM symposium on Software ..., 2003 - portal.acm.org
... calculations due to the flatness of the inheritance **hierarchy**. ... also draw attention to the **flow** of control ... Figure 8. **Object** Diagram with focus + context applied ...
Cited by 10 - Related Articles - Web Search

Hpctoolkit: Multi-platform tools for profile-based performance **analysis**
J Mellor-Crummey - 5th International Workshop on Automatic Performance **Analysis** ..., 2003 - cs.rice.edu
... complex memory **hierarchy** non-blocking, multi-level caches TLB ... binary **object** code compilation ... Construct control **flow** graph using branch target **analysis** ...
Cited by 4 - Related Articles - View as HTML - Web Search

Gooooooooogle ▶
Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

object hierarchy flow hierarchy annc Search

Google Home - About Google - About Google Scholar

©2007 Google

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((visualizer graph control flow annotate hierarchy object) <in>metadata)) <and> (pyr >..."
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((visualizer graph control flow annotate hierarchy ob

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

» Key

- **IEEE JNL** IEEE Journal or Magazine
- **IEE JNL** IEE Journal or Magazine
- **IEEE CNF** IEEE Conference Proceeding
- **IEE CNF** IEE Conference Proceeding
- **IEEE STD** IEEE Standard

Indexed by Inspec

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((graph based visualization)<in>metadata)) <and> (pyr >= pyr <= ...)"
Your search matched 2 of 1450046 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevanc Descending order.

» Search Options

View Session History
New Search

Modify Search

(((graph based visualization)<in>metadata)) <and> (

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

view selected items Select All Deselect All

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

☐ 1. **DataViewer: A scene graph based visua**
Paffenroth, R.; Vrajitoru, D.;
Eurographics UK Conference, 2002. Proc
11-13 June 2002 Page(s):147 - 148
Digital Object Identifier 10.1109/EGUK.2
AbstractPlus | Full Text: PDF(230 KB)
Rights and Permissions

☐ 2. **Fuzzy queries and cross-language ontol exploitation**
Cross, V.V.; Voss, C.R.;
Fuzzy Systems, 2000. FUZZ IEEE 2000.
Conference on
Volume 2, 7-10 May 2000 Page(s):641 -
Digital Object Identifier 10.1109/FUZZY.
AbstractPlus | Full Text: PDF(264 KB)
Rights and Permissions

Indexed by Inspec